

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2000-306076

(43)Date of publication of application : 02.11.2000

(51)Int.Cl. G06T 1/00
G06F 12/00
H04N 1/00

(21)Application number : 11-112826

(71)Applicant : CANON INC

(22)Date of filing : 20.04.1999

(72)Inventor : HISAGAI MASAMI

(54) IMAGE PROCESSOR, CONTROL METHOD AND STORAGE MEDIUM

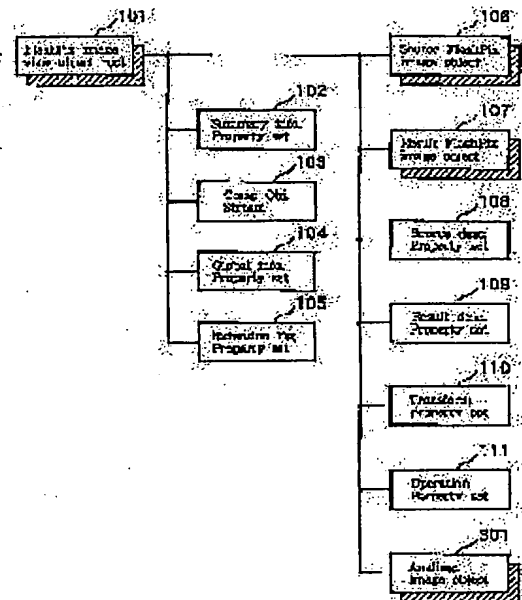
(57)Abstract:

PROBLEM TO BE SOLVED: To enable not only the retrieval of document images but also the provision of images with a desired definition.

SOLUTION: Character information and non-character information provided by inputting and analyzing an image including sentences are stored and managed.

At such a time, concerning an image part to be a non-character area, the image data of a hierarchical structure with different resolution are generated.

Then, the generated image data of the hierarchical structure are preserved in the unit of character information. Concerning the character information, an analyzed image object 301 is further provided in a flash pixel format and it is preserved here.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (JP) (12) 公開特許公報 (A) (11) 特許出願公開番号

特開2000-306076
(P.2000-306076A)
(43) 公開日 平成12年11月2日 (2000.11.2)

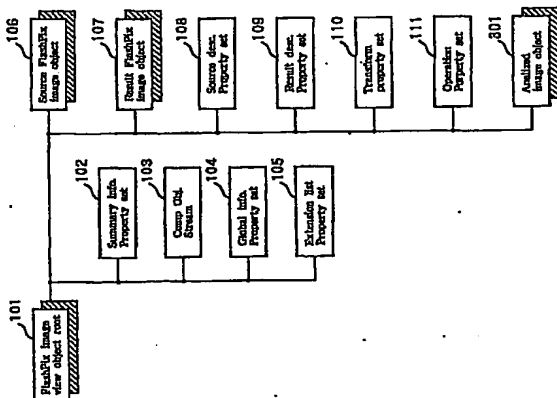
(51) Int. Cl. ⁷	識別記号	F I	テーマコード (参考)
G 0 6 F	1/00	G 0 6 F	15/62 3 3 0 A 58050
G 0 6 F	12/00	12/00	5 2 0 J 58032
H 0 4 N	1/00	H 0 4 N	1/00 1 0 7 Z 5C062

審査請求	未請求	請求項の数	OL	(全10頁)
(21) 出願番号	特願平11-112326	(71) 出願人	000001007	
(22) 出願日	平成11年4月20日 (1999.4.20)	キヤノン株式会社		
		東京都大田区下丸子3丁目30番2号		
		久保 正己		
		東京都大田区下丸子3丁目30番2号		
		ン株式会社内		
		100076428		
		弁理士 大塚 康徳 (外2名)		

最終頁に続く

(54) 【発明の名称】 画像処理装置及び制御方法及び記憶媒体

(57) 【要約】
【課題】 文書画像の読解は勿論、要望される品位の画像を提供する。
【解決手段】 文章を含む画像を入力し、解析して得られた文字情報及び非文字情報を記憶管理する。このとき、非文字領域となる画像部分については解像度の異なる階層構造の画像データを生成する。そして、生成された階層構造の画像データを前記文字情報とを単位として保存する。文字情報については、フラッシュメモリーに保存する。



【特許請求の範囲】

【請求項1】 文字列を含む画像データを解析して得られた文字情報及び非文字情報を記憶管理する文書画像ファイルに記憶装置であって、

非文字情報については解像度の異なる階層構造の画像データを生成する画像生成手段と、

生成された階層構造の画像データを前記文字情報とを単位として保存する保存手段とを備えることを特徴とする画像処理装置。

【請求項2】 前記保存手段で保存するファイル形式はフラッシュメモリーに記憶の画像処理装置。

【請求項3】 前記保存手段で保存する情報には、保存対象のデータにリンクするためのアドレス情報が含まれることを特徴とする請求項1項に記載の画像処理装置。

【請求項4】 更に、ネットワークを介して通信する通信手段を備え、前記保存手段で保存されたデータを当該通信手段によって送受信することとを特徴とする請求項1項に記載の画像処理装置。

【請求項5】 文字列を含む画像データを解析して得られた文字情報及び非文字情報を記憶管理する画像処理装置の制御方法であって、

非文字情報については解像度の異なる階層構造の画像データを生成する画像生成工程と、

生成された階層構造の画像データを前記文字情報とを単位として保存する保存工程とを備えることを特徴とする画像処理装置の制御方法。

【請求項6】 コンピュータが読み込み実行すること、文字列を含む画像データを解析して得られた文字情報及び非文字情報を記憶管理する文書処理装置として機能する、プログラムコードを格納した記憶媒体であって、

非文字情報については解像度の異なる階層構造の画像データを生成する画像生成手段と、

生成された階層構造の画像データを前記文字情報とを単位として保存する保存手段として機能するプログラムコードを格納した記憶媒体。

【発明の詳細な説明】

【0001】
【発明の属する技術分野】 本発明は文書画像データをファイルリングする画像処理装置及びその制御方法及び記憶媒体に関するものである。

【0002】
【従来の技術】 文書画像データになんらかの文書解析処理を施した結果のデータをファイルリング装置がある。また、この装置をネットワーク等を介して遠方に送信することも行われている。ファイルリングする場合には、文書画像中の文章部分をOCRしてテキストデータを文書画像とともに記憶したり、あるいは遠隔地に送信したりして、文字列による処理（たとえば検索）を行う場合に

利用できる。

【0003】 また、モノクロ文書画像を、その画像中のOCR結果のテキストを付随させて、全文検索のできるファイルリングシステムの技術も存在する。

【0004】 一方、FlashPixと呼ばれる画像ファイルのフォーマットの標準化が検討されている。この画像フォーマットのファイルは、一つの画像について複数の階層のデータが含まれるため、画像のサムネイル表示、ブローゼリングな画像表示などを可能としている。

【0005】

【発明が解決しようとする課題】 しかしながら、前記全文検索のできるファイルリングシステムの技術では、ひとつの文書画像の性質は固定的に一種類のみであり、OCRなどの処理をすることが煩雑の場合、文書画像はOCRに適用したモノクロ2値画像で、解像度が300～400dpiが条件になるため、文書画像はその条件で記憶されたり、送信されることが制限されていた。一方、文書画像を画像ビューアで見ると、高くて200dpi程度のカラー画像が見え、また、カラー画像dpi程度のカラールームがあるため、サムネイル画像を一覧表示したり、拡大解像度で見ると、低い解像度から高い解像度へブローゼリングに表示することが必要であった。そのため、OCRと画像処理の性質と矛盾するため、OCRと画像処理とは両立できなかつた。

【0006】 また、"FlashPix format and Architecture White Paper", June 17, 1996, Eastman Kodak Company. によれば、標準化案となっているFlashPix formatには、OCRなどの文 解析結果のデータが含まれないため、テキストによる検索など、テキストベースの処理ができない。

【0007】 本発明はかかる問題点に鑑み、なされたものであり、文書画像の読解は勿論、要望される品位の画像を提供することを可能ならしめる画像処理装置及びその制御方法及び記憶媒体を提供しようとするものである。

【0008】

【課題を解決するための手段】 この課題を解決するため、例えば本発明の画像処理装置は以下に示す構成を備える。すなわち、文字列を含む画像データを解析して得られた文字情報及び非文字情報を記憶管理する文書画像ファイルに記憶装置であって、非文字情報については解像度の異なる階層構造の画像データを生成する画像生成手段と、生成された階層構造の画像データを前記文字情報とを単位として保存する保存手段とを備える。

【0009】 また、本発明の好適な実施形態に依れば、本発明は、画像データと、該画像データから生成される解像度を有した第二の画像データと、さらに第二の画像データから生成される解像度を有した第三の画像データとを含み、一般に第nの画像データから階層的に生成される解像度を有した第(n+1)の画像データを

含む複数の画像データを含む文書画像ファイルであって、前記画像データを文書解析した結果の文書解析データを含むことを特徴とする文書画像ファイルを記述するメタデータ、または該文書画像データを通過経路を通して送受信する方法を提供する。

【0010】また、前記文書解析データには、前記画像データに関連する情報をリンクするためのアドレス情報を含むことを特徴とする。

【0011】

【発明の実施の形態】以下、添付図面に従って本発明に係る実施形態を詳細に説明する。

【0012】図1と図2は、FlashPix formatファイルの説明する図であり、先ず、これについて説明する。

【0013】ファイル内の各プロパティやデータには、MS-DOS（米国Microsoft社の商標）のディレクトリとファイルに相当する。ストレージとストリームによってアクセスする。図1、図2において、影付き部分がストレージを示し、影なし部分がストリームである。画像データや画像属性情報はストリーム部分に格納され、画像データは異なる解像度で階層化されており、それぞれその解像度の画像をSubImageと呼び、Resolution 0〜nで示してある。各解像度画像に対して、その画像を呼び出すために必要な情報がSubImage headerに、また画像データがSubImage dataに格納される。プロパティセットとは属性情報をその使用目的、内容に応じて分類して定義したもので、Summary Info、Property Set、Image Info、Property Set、Image Content Property Set、Extension list property Setがある。

【0014】[各プロパティの説明] Summary Info、Property Setは、FlashPix特有のものではなく、米国Microsoft社のストラクチャードストレージでは必須のプロパティセットで、そのファイルのタイトル・題名・著者・サムネイル画像等を格納する。

【0015】Image Content Property Setは、画像データの格納方法を記述する属性である。この属性には画像データの階層数、最大解像度の画像の幅、高さ、それぞれの解像度の画像についての幅、高さ、色の構成、あるいはJPEG圧縮を用いる際の量子化テーブル・ハフマンテーブルの定義を記述する。

【0016】Image Info、Property Setは、画像を使用する際に利用できる様々な情報、たとえば、画像がどのようなようにして取り込まれ、どのように利用可能であるかの情報を格納する。

【0017】デジタルデータの取込み方法/使いは生成方法に関する情報 (File Source)

・画像の内容 (画像中の人物、場所など) に関する情報 (Content description)

・撮影に使われたカメラに関する情報 (Camera information)

・撮影時のカメラのセッティング (露出、シャッタース

peed、絞り、白平衡、フラッシュ使用の有無など) の情報 (Per Picture camera settings)

・デジタルカメラ特有の解像度やモザイクフィルタに関する情報 (Digital camera characterization)

・フィルムカメラのメタデータ、製品名、種類 (ネガ/ポジ、カラー/白黒) などの情報 (Film description)

・オリジナルが写物や印刷物である場合の題名やサイズに関する情報 (Original document scan description)

・スキャン画像の場合、使用したスキャナやソフト、操作した人に関する情報 (Scan device)

Extension list property Setは上記FlashPixの基本仕様に含まれない情報を追加する際に使用する領域である。

【0017】図1の101〜111において、Flash Image View Objectは、画像を表示する際に用いるビューイングパラメータと画像データを合わせて格納する画像ファイルである。ビューイングパラメータとは画像の回転、拡大/縮小、移動、色変換、フィルタリングの処理を画像表示の際に適用するために記憶しておく処理情報のセットである。

【0018】Source/Result FlashPix Image ObjectはFlashPix画像データの本体であり、Source FlashPix Image Objectは必須、Result FlashPix Image Objectはオプションである。

【0019】Source FlashPix Image ObjectはOriginalの画像データを、Result FlashPix Image Objectはビューイングパラメータを使い画像処理した結果の画像を格納する。

【0020】Source/Result desc. Property setは上記、画像データの識別のためのプロパティセットであり、画像ID、変更禁止のプロパティセット、最終変更日時を格納する。

【0021】Transform property setは回転、拡大/縮小、移動のためのAffine変換係数、色変換マトリクス、コントラスト調整値、フィルタリング係数を格納している。次に画像データの取り扱いについて説明する。

【0022】図4に解像度の異なる複数の画像から構成される画像ファイルの例を示す。図4で最大解像度の画像がC×Rの画素で構成されており、その次に大きい画像はC/2×C/2であり、それ以降画素数、行数ともに1/2ずつ縮小し、列・行ともに64画素以下あるいは等しくなるまで繰り返す。このように階層化した結果、画像の属性情報として「1つの画像ファイル中の階層数」やそれぞれの階層の画像に対して、ヘッダ情報と画像データが必要となる。1つの画像ファイル中の階層の数や最大解像度の画像の幅、高さ、あるいはそれぞれの解像度の画像の幅、高さ、色構成、圧縮方式等に関する情報は前記Image Contents Property Setに記述される。

【0023】さらに各解像度のレイヤの画像は64×64ピクセル単位で構成されている。図5の例では「5」（ブロック601〜606）である。Block DATAには、次の格納体 (DATA) が先頭に配置されている。

struct tagBdata {
int x1; //ブロックの左上角のx座標
int y1; //ブロックの左上角のy座標
int x2; //ブロックの右下角のx座標
int y2; //ブロックの右下角のy座標
int kind; //ブロックの種類
int offset; //ブロックデータのオフセット
} BDATA;

ここで、ブロックの種類kindは、たとえば次のように定義される。

【0028】
0: テキスト
1: 表
2: 線画

【0029】ところで、この文書解析データ (Unalised image object) は、FlashPix ImageObjectの通常な解像度の画像を、待機平06-068301号公報に開示された方法で作成される。あるいは、FlashPix image objectには無い他の解像度でスキャンされた画像から作成されてもかまわない。FlashPix image objectを作成するための適当なアプリケーションソフトウェアが必要な

【0025】たとえば、図6のように、文書画像601から、602、603、604、605、606の各ブロックが抽出され、該ブロックの位置・大きさ・種類が求められる。ブロック602は見出しテキストブロック、ブロック603および604は本文テキストブロック、ブロック605、606はビジュアルブロックである。

【0026】また、図7は、Analised image objectの内容、すなわち文書解析データ (Block Information (501)、Text Data (502)、Key Data (503)) の表現方法の一例を示したものである。701は、Headerであり、その内容をC言語で表すと、

struct header {
int BInfo_offset; //Block Information dataのオフセット
int BInfo_dataLength; //Block Information dataのデータ長 (単位: バイト)
int BData_offset; //Block Dataの先頭のオフセット
int BData_dataLength; //Block Dataのデータ長 (単位: バイト)
int KeyData_offset; //Key Dataの先頭のオフセット
int KeyData_dataLength; //Key Dataのデータ長 (単位: バイト)
} header;

【0027】図7で、Block numberは、文書に含まれる領域の個数であり、図6の例では「5」（ブロック601〜606）である。Block DATAには、次の格納体 (DATA) が先頭に配置されている。

struct tagBdata {
int x1; //ブロックの左上角のx座標
int y1; //ブロックの左上角のy座標
int x2; //ブロックの右下角のx座標
int y2; //ブロックの右下角のy座標
int kind; //ブロックの種類
int offset; //ブロックデータのオフセット
} BDATA;

ここで、ブロックの種類kindは、たとえば次のように定義される。

【0028】
0: テキスト
1: 表
2: 線画

【0029】ところで、この文書解析データ (Unalised image object) は、FlashPix ImageObjectの通常な解像度の画像を、待機平06-068301号公報に開示された方法で作成される。あるいは、FlashPix image objectには無い他の解像度でスキャンされた画像から作成されてもかまわない。FlashPix image objectを作成するための適当なアプリケーションソフトウェアが必要な

【0025】たとえば、図6のように、文書画像601から、602、603、604、605、606の各ブロックが抽出され、該ブロックの位置・大きさ・種類が求められる。ブロック602は見出しテキストブロック、ブロック603および604は本文テキストブロック、ブロック605、606はビジュアルブロックである。

【0026】また、図7は、Analised image objectの内容、すなわち文書解析データ (Block Information (501)、Text Data (502)、Key Data (503)) の表現方法の一例を示したものである。701は、Headerであり、その内容をC言語で表すと、

struct header {
int BInfo_offset; //Block Information dataのオフセット
int BInfo_dataLength; //Block Information dataのデータ長 (単位: バイト)
int BData_offset; //Block Dataの先頭のオフセット
int BData_dataLength; //Block Dataのデータ長 (単位: バイト)
int KeyData_offset; //Key Dataの先頭のオフセット
int KeyData_dataLength; //Key Dataのデータ長 (単位: バイト)
} header;

【0027】図7で、Block numberは、文書に含まれる領域の個数であり、図6の例では「5」（ブロック601〜606）である。Block DATAには、次の格納体 (DATA) が先頭に配置されている。

struct tagBdata {
int x1; //ブロックの左上角のx座標
int y1; //ブロックの左上角のy座標
int x2; //ブロックの右下角のx座標
int y2; //ブロックの右下角のy座標
int kind; //ブロックの種類
int offset; //ブロックデータのオフセット
} BDATA;

ここで、ブロックの種類kindは、たとえば次のように定義される。

【0028】
0: テキスト
1: 表
2: 線画

【0029】ところで、この文書解析データ (Unalised image object) は、FlashPix ImageObjectの通常な解像度の画像を、待機平06-068301号公報に開示された方法で作成される。あるいは、FlashPix image objectには無い他の解像度でスキャンされた画像から作成されてもかまわない。FlashPix image objectを作成するための適当なアプリケーションソフトウェアが必要な

【0025】たとえば、図6のように、文書画像601から、602、603、604、605、606の各ブロックが抽出され、該ブロックの位置・大きさ・種類が求められる。ブロック602は見出しテキストブロック、ブロック603および604は本文テキストブロック、ブロック605、606はビジュアルブロックである。

【0026】また、図7は、Analised image objectの内容、すなわち文書解析データ (Block Information (501)、Text Data (502)、Key Data (503)) の表現方法の一例を示したものである。701は、Headerであり、その内容をC言語で表すと、

struct header {
int BInfo_offset; //Block Information dataのオフセット
int BInfo_dataLength; //Block Information dataのデータ長 (単位: バイト)
int BData_offset; //Block Dataの先頭のオフセット
int BData_dataLength; //Block Dataのデータ長 (単位: バイト)
int KeyData_offset; //Key Dataの先頭のオフセット
int KeyData_dataLength; //Key Dataのデータ長 (単位: バイト)
} header;

【0027】図7で、Block numberは、文書に含まれる領域の個数であり、図6の例では「5」（ブロック601〜606）である。Block DATAには、次の格納体 (DATA) が先頭に配置されている。

4画素のタイルに分割されている。画像の左上から順次64×64のタイルに分割すると、画像によっては右端および左端の一部に空白が生ずる場合がある。この場合はそれぞれ左端および右端の画像を繰り返し挿入すること、64×64の画像を格納する。FlashPix画像では、それぞれのタイル中の画像をJPEG圧縮、シグナルカラー、非圧縮のいずれかの方法で格納する。JPEG圧縮は、ISO/IEC、JTC1/SC29により、方式自体の説明はここでは割愛する。このようにタイル分割された画像データは、SubImage dataストリーム中に格納され、タイルの種別、個々のタイルのサイズ、データの開始位置、圧縮方法はすべてSubImage headerに格納されている。シングルカラーとは、前記1つのタイルがすべて同じ色で構成されている場合にのみ、個々の画素の値を記述することなく、そのタイルの色を1色で表現する方式である。この方法は特にコンピュータグラフィックスにより生成された画像に対して有効である。

【0024】図3は実施形態の特徴を示すものであり、FlashPix image view object rootに、Analised image objectを追加したものである。Analised image objectは、文書解析データであり、図5のようにBlock Information (501)、Text Data (502)、Key Data (503) から構成 *

【0025】たとえば、図6のように、文書画像601から、602、603、604、605、606の各ブロックが抽出され、該ブロックの位置・大きさ・種類が求められる。ブロック602は見出しテキストブロック、ブロック603および604は本文テキストブロック、ブロック605、606はビジュアルブロックである。

【0026】また、図7は、Analised image objectの内容、すなわち文書解析データ (Block Information (501)、Text Data (502)、Key Data (503)) の表現方法の一例を示したものである。701は、Headerであり、その内容をC言語で表すと、

struct header {
int BInfo_offset; //Block Information dataのオフセット
int BInfo_dataLength; //Block Information dataのデータ長 (単位: バイト)
int BData_offset; //Block Dataの先頭のオフセット
int BData_dataLength; //Block Dataのデータ長 (単位: バイト)
int KeyData_offset; //Key Dataの先頭のオフセット
int KeyData_dataLength; //Key Dataのデータ長 (単位: バイト)
} header;

【0027】図7で、Block numberは、文書に含まれる領域の個数であり、図6の例では「5」（ブロック601〜606）である。Block DATAには、次の格納体 (DATA) が先頭に配置されている。

struct tagBdata {
int x1; //ブロックの左上角のx座標
int y1; //ブロックの左上角のy座標
int x2; //ブロックの右下角のx座標
int y2; //ブロックの右下角のy座標
int kind; //ブロックの種類
int offset; //ブロックデータのオフセット
} BDATA;

ここで、ブロックの種類kindは、たとえば次のように定義される。

【0028】
0: テキスト
1: 表
2: 線画

【0029】ところで、この文書解析データ (Unalised image object) は、FlashPix ImageObjectの通常な解像度の画像を、待機平06-068301号公報に開示された方法で作成される。あるいは、FlashPix image objectには無い他の解像度でスキャンされた画像から作成されてもかまわない。FlashPix image objectを作成するための適当なアプリケーションソフトウェアが必要な

【0025】たとえば、図6のように、文書画像601から、602、603、604、605、606の各ブロックが抽出され、該ブロックの位置・大きさ・種類が求められる。ブロック602は見出しテキストブロック、ブロック603および604は本文テキストブロック、ブロック605、606はビジュアルブロックである。

【0026】また、図7は、Analised image objectの内容、すなわち文書解析データ (Block Information (501)、Text Data (502)、Key Data (503)) の表現方法の一例を示したものである。701は、Headerであり、その内容をC言語で表すと、

struct header {
int BInfo_offset; //Block Information dataのオフセット
int BInfo_dataLength; //Block Information dataのデータ長 (単位: バイト)
int BData_offset; //Block Dataの先頭のオフセット
int BData_dataLength; //Block Dataのデータ長 (単位: バイト)
int KeyData_offset; //Key Dataの先頭のオフセット
int KeyData_dataLength; //Key Dataのデータ長 (単位: バイト)
} header;

【0027】図7で、Block numberは、文書に含まれる領域の個数であり、図6の例では「5」（ブロック601〜606）である。Block DATAには、次の格納体 (DATA) が先頭に配置されている。

struct tagBdata {
int x1; //ブロックの左上角のx座標
int y1; //ブロックの左上角のy座標
int x2; //ブロックの右下角のx座標
int y2; //ブロックの右下角のy座標
int kind; //ブロックの種類
int offset; //ブロックデータのオフセット
} BDATA;

ここで、ブロックの種類kindは、たとえば次のように定義される。

【0028】
0: テキスト
1: 表
2: 線画

【0029】ところで、この文書解析データ (Unalised image object) は、FlashPix ImageObjectの通常な解像度の画像を、待機平06-068301号公報に開示された方法で作成される。あるいは、FlashPix image objectには無い他の解像度でスキャンされた画像から作成されてもかまわない。FlashPix image objectを作成するための適当なアプリケーションソフトウェアが必要な

【0025】たとえば、図6のように、文書画像601から、602、603、604、605、606の各ブロックが抽出され、該ブロックの位置・大きさ・種類が求められる。ブロック602は見出しテキストブロック、ブロック603および604は本文テキストブロック、ブロック605、606はビジュアルブロックである。

【0026】また、図7は、Analised image objectの内容、すなわち文書解析データ (Block Information (501)、Text Data (502)、Key Data (503)) の表現方法の一例を示したものである。701は、Headerであり、その内容をC言語で表すと、

struct header {
int BInfo_offset; //Block Information dataのオフセット
int BInfo_dataLength; //Block Information dataのデータ長 (単位: バイト)
int BData_offset; //Block Dataの先頭のオフセット
int BData_dataLength; //Block Dataのデータ長 (単位: バイト)
int KeyData_offset; //Key Dataの先頭のオフセット
int KeyData_dataLength; //Key Dataのデータ長 (単位: バイト)
} header;

【0027】図7で、Block numberは、文書に含まれる領域の個数であり、図6の例では「5」（ブロック601〜606）である。Block DATAには、次の格納体 (DATA) が先頭に配置されている。

struct tagBdata {
int x1; //ブロックの左上角のx座標
int y1; //ブロックの左上角のy座標
int x2; //ブロックの右下角のx座標
int y2; //ブロックの右下角のy座標
int kind; //ブロックの種類
int offset; //ブロックデータのオフセット
} BDATA;

ここで、ブロックの種類kindは、たとえば次のように定義される。

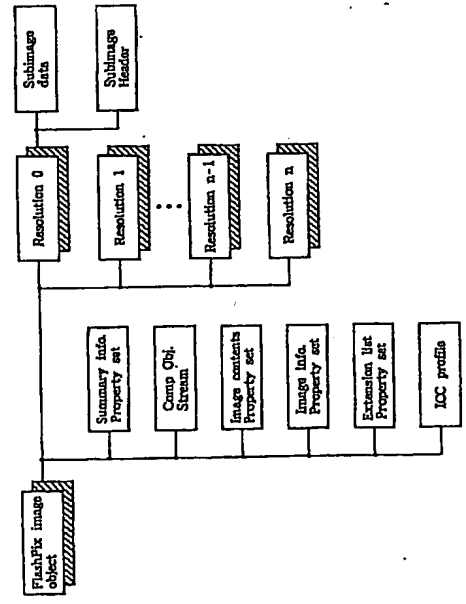
【0028】
0: テキスト
1: 表
2: 線画

【0029】ところで、この文書解析データ (Unalised image object) は、FlashPix ImageObjectの通常な解像度の画像を、待機平06-068301号公報に開示された方法で作成される。あるいは、FlashPix image objectには無い他の解像度でスキャンされた画像から作成されてもかまわない。FlashPix image objectを作成するための適当なアプリケーションソフトウェアが必要な

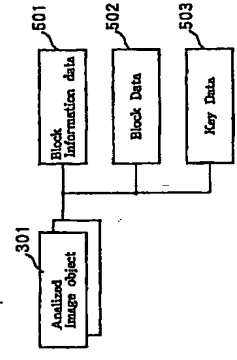
【0025】たとえば、図6のように、文書画像601から、602、603、604、605、606の各ブロックが抽出され、該ブロックの位置・大きさ・種類が求められる。ブロック602は見出しテキストブロック、ブロック603および604は本文テキストブロック、ブロック605、606はビジュアルブロックである。

【0026】また、図7は、Analised image objectの内容、すなわち文書解析データ (Block Information (501)、Text Data (502)、Key Data (503)) の表現方法の一例を示したものである。701は、Headerであり、その内容をC言語で表すと、

[圖 2]



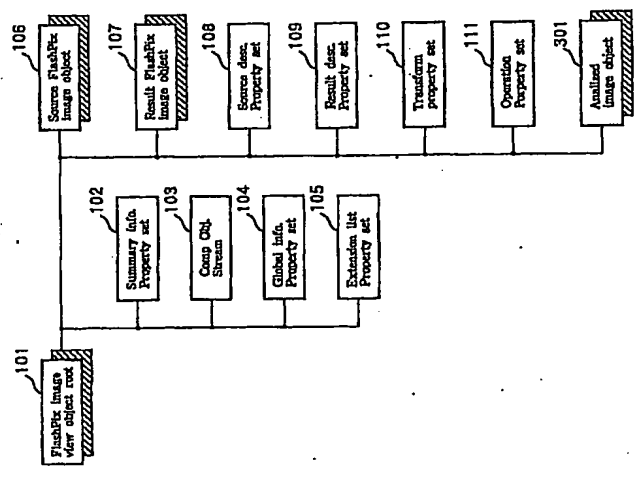
[圖 5]



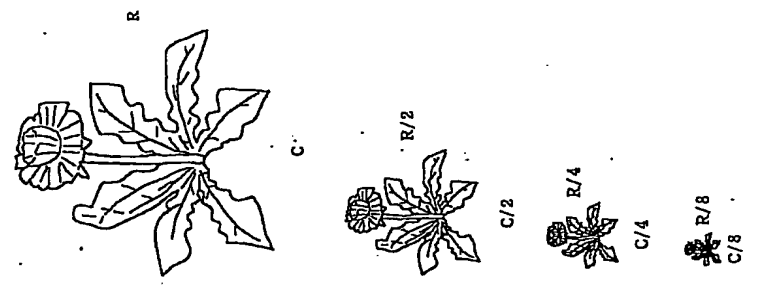
[圖 7]

Header	701
Block number	702
Block information data (1)	703
...	...
Block information data (n)	704
Block Data (1)	705
Block Data (n)	706
...	...
Block Data (n)	707
Key Data	708

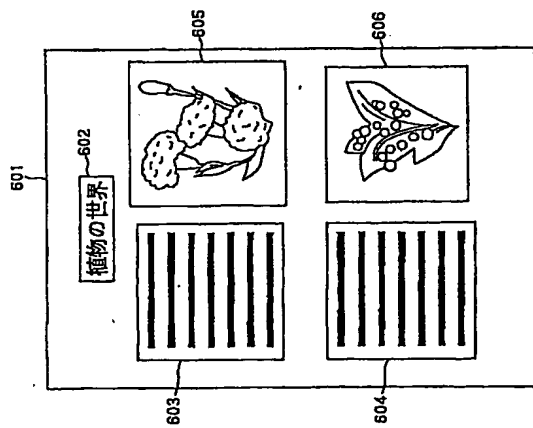
[圖 3]



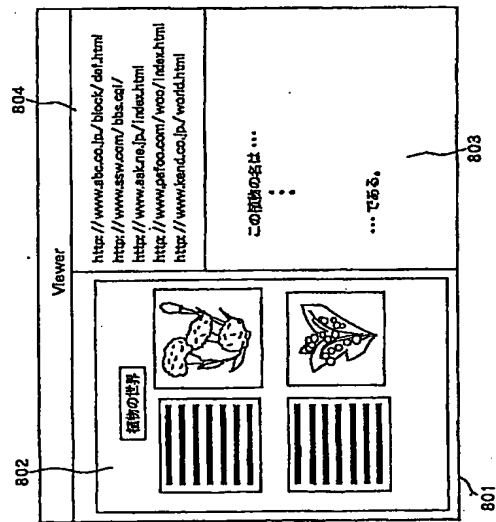
[圖 4]



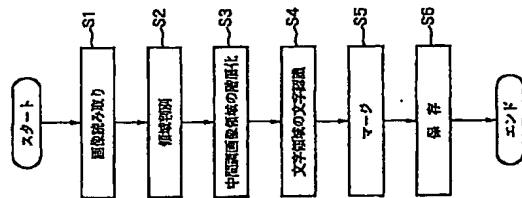
【図6】



【図8】



【図10】



フロントページの続き

Fターム(参考) 58050 BA06 BA10 BA16 CA04 CA07
DA06 EA03 EA05 EA06 EA10
EA12 EA19 FA02 FA12 FA19
CA08
58082 AA13 EA01
5C062 AA06 AB17 AB23 AB38 AB42
AC08 AC22 AC24 AC51 AF00
BA00